

Claims

We claim:

1 1. A method for training a self ordering map,
2 comprising the steps of:

3 initializing a set of weights of a self-ordering map;
4 iteratively training said weights over many training
5 epochs;

6 for at least a number of said epochs, said step of
7 iteratively training including updating said weights based
8 on a learning rate that is generated according to a
9 function that changes in a fashion that is other than
10 monotonically a decreasing value with training epoch.

1 2. A method as in claim 1, wherein said step of
2 iteratively training includes updating said weights based
3 on a learning rate that is generated according to a random
4 or pseudorandom function.

1 3. A method as in claim 2 wherein said step of
2 iteratively training includes updating said weights based
3 on a learning rate that is generated according to a
4 function that is such that values over which said learning
5 rate may range decreases with training epoch.

1 4. A method as in claim 2 wherein said step of
2 iteratively training includes updating said weights based

3 on a learning rate that is generated according to a
4 function that is such that values over which said learning
5 rate tend to decrease with training epoch.

1 5. A method as in claim 1 wherein said step of
2 iteratively training includes updating said weights based
3 on a learning rate that is generated according to a
4 function that is such that values over which said learning
5 rate may range decreases with training epoch.

1 6. A method as in claim 5 wherein said step of
2 iteratively training includes updating said weights based
3 on a learning rate that is generated according to a
4 function that is such that values over which said learning
5 rate tend to decrease with training epoch.

1 7. A method as in claim 1 wherein said step of
2 iteratively training includes updating said weights based
3 on a learning rate that is generated according to a
4 function that is such that values over which said learning
5 rate tend to decrease with training epoch.

1 8. A method of training a self ordering feature map,
2 comprising the steps of:
3 choosing a random value for initial weight vectors;

4 drawing a sample from a set of training sample vectors
5 and applying it to input nodes of said self ordering
6 feature map;

7 identifying a winning competition node of said self
8 ordering feature map according to a least distance
9 criterion;

10 adjusting a synaptic weight of at least said winning
11 node;

12 said step of adjusting including selecting a value for
13 a learning rate used to update said synaptic weight that is
14 based on a function other than one that is monotonic with
15 training epoch;

16 iteratively repeating said steps of drawing,
17 identifying, and adjusting.

1 9. A method as in claim 8, wherein said step of
2 adjusting includes updating said weights based on a
3 learning rate that is generated according to a random or
4 pseudorandom function.

1 10. A method as in claim 9 wherein said step of
2 adjusting includes updating said weights based on a
3 learning rate that is generated according to a function
4 that is such that values over which said learning rate may
5 range decreases with training epoch.

1 11. A method as in claim 9 wherein said step of
2 adjusting includes updating said weights based on a
3 learning rate that is generated according to a function
4 that is such that values over which said learning rate tend
5 to decrease with training epoch.

1 12. A method as in claim 8 wherein said step of
2 adjusting includes updating said weights based on a
3 learning rate that is generated according to a function
4 that is such that values over which said learning rate may
5 range decreases with training epoch.

1 13. A method as in claim 12 wherein said step of
2 adjusting includes updating said weights based on a
3 learning rate that is generated according to a function
4 that is such that values over which said learning rate tend
5 to decrease with training epoch.

1 14. A method as in claim 8 wherein said step of
2 adjusting includes updating said weights based on a
3 learning rate that is generated according to a function
4 that is such that values over which said learning rate tend
5 to decrease with training epoch.